J. Symbolic Computation (1997) **24**, 409–432

A New Algorithm for the Computation of Canonical Forms of Matrices over Fields

ALLAN STEEL†

School of Mathematics and Statistics,
University of Sydney,
Sydney NSW 2006,
Australia

A new algorithm is presented for the computation of canonical forms of matrices over fields. These are the *Primary Rational*, *Rational*, and *Jordan* canonical forms. The algorithm works by obtaining a decomposition of the vector space acted on by the given matrix into *primary* cyclic spaces (spaces whose minimal polynomials with respect to the matrix are powers of irreducible polynomials). An efficient implementation of the algorithm is incorporated in the MAGMA Computer Algebra System.

1. Introduction

Computing canonical forms of matrices over fields is a classical mathematical problem with many applications in all areas of mathematics. A natural way in which these forms arise is in considering the action of an element of a matrix algebra on its underlying vector space. This leads to the notion of a subspace which is invariant under the action. By decomposing the vector space into a direct sum of *cyclic* subspaces under this action, the structure of the action can be understood. In terms of matrices, finding a basis for the subspaces in this decomposition leads to a transformation matrix T such that when the original matrix X is conjugated by T, a matrix is obtained with a simple structure. The matrices with the simple structure are called *canonical forms*.

The most common canonical forms encountered in the literature are the *Rational* and *Jordan* forms. The *Rational* form of a square matrix over a field always exists and is unique. The *Jordan* form of a square matrix usually only exists over an algebraically closed field. In this paper we define also the *Primary Rational* form and the *generalized Jordan* form, both of which always exist over any field. The generalized Jordan form of a matrix does equal the usual Jordan form when computed over an algebraically closed field.

The algorithm presented in this paper was designed to be implemented in the MAGMA Computer Algebra System (Bosma *et al.*, 1997). That system contains a new efficient

† E-mail: allan@maths.usyd.edu.au

0747–7171/97/030409 + 24 $18.00/0 sy960142

linear algebra package which concentrates on efficient packed representations of vectors (particularly over finite fields) so that row operations on matrices are especially efficient. This means that, for example, if the canonical form of a matrix over a small finite field is desired, it is important that the algorithm takes advantage of these representations so that most of the computation actually consists of row operations on such matrices and other operations like column operations and matrix operations over larger rings are avoided.

One well-known algorithm for computing the Rational form is the method of Danilevsky (Faddeeva, 1959). This algorithm has the limitation in the current context that it depends heavily on column operations as well as row operations. Also, the algorithm does not calculate the Jordan form if that is desired. Other algorithms for computing the Rational form are given in Howell (1973) (which works by modular arithmetic), and Mathieu and Ford (1990) (which works by p-adic approximation). For an algorithm to compute the true Jordan form of a matrix (assuming the characteristic polynomial factorizes into linear factors), see Dixon *et al.* (1990).

The new algorithm presented in this paper computes all of the above canonical forms by the method of "spinning vectors"—that is, simply multiplying vectors from the vector space by the given matrix to compute subspaces which are invariant under the action of that matrix. This needs row operations only—no column operations.

The algorithm also makes large use of polynomial arithmetic. In particular, the factorization of univariate polynomials over the given field plays a large part. This is not surprising in that the structure of the generalized Jordan form contains information yielding the factorization of the characteristic polynomial of the given matrix. The algorithm actually works by decomposing the underlying vector space into a direct sum of invariant subspaces such that the minimal polynomials of the given matrix acting on these subspaces are *powers of irreducible polynomials*; such subspaces are called *primary*. This property enables the computation of *cyclic generators* of the subspaces which yield a transformation matrix to conjugate the given matrix into the canonical form.

2. Mathematical Basics

Let K be a field, V the vector space $K^{(n)}$ of n-tuples over K, and fix a matrix $X \in K^{(n\times n)}$.

We first define some notation commonly used in the rest of the paper: for a polynomial $p(x) \in K[x]$, we let $\partial p(x)$ denote the degree of $p(x)$; for a subset U of V we let $U \leq V$ denote that U is a subspace of V and we let $U < V$ denote that U is a *proper* subspace of V; and for a subset S of V we let $\langle S\rangle$ denote the subspace of V generated by S.

Using the simple multiplication on the right of a (row) vector by a matrix, X acts on V, and V can be considered as a $K[x]$-module via X by the multiplication definition

$$v \cdot f(x) = v \cdot f(X)$$

for $v \in V$ and $f(x) \in K[x]$.

Definition 2.1. *For $v \in V$, we define* $\mathrm{Orb}_X(v)$, *the* orbit of v under X, *to be the set*

$$\{v \cdot f(X) :\ f(x) \in K[x]\}.$$

Clearly $\mathrm{Orb}_X(v)$ *is an X-invariant subspace of V.* $\mathrm{Orb}_X(v)$ *is called the subspace of V*

(acted on by X) cyclically generated *by v. Also, an X-invariant subspace of V is called* cyclic *if it is the orbit under X of some vector from V.*

Definition 2.2. *For $v \in V$, we define* $\mathrm{Min}_X(v)(x)$, *the* minimal polynomial of v with respect to X, *to be the smallest-degree non-zero monic polynomial $f(x) \in K[x]$ such that $v \cdot f(X) = 0$. It is easy to see that such an $f(x)$ always exists and is unique.*

Definition 2.3. *For an X-invariant subspace U of V, we define* $\mathrm{Min}_X(U)(x)$ *to be the smallest degree monic non-zero polynomial $f(x) \in K[x]$ such that $u.f(X) = 0$ for all $u \in U$. Again it is easy to see that such an $f(x)$ always exists and is unique.*

We now quote two important theorems which assert that the vector space V acted on by X can be decomposed into cyclic subspaces. This leads to the existence of the canonical forms discussed in this paper.

Theorem 2.4. (Hartley and Hawkes (1970), 8.2) *Let K be a field, $V = K^{(n)}$ and fix $X \in K^{(n\times n)}$. Then there exists $k \in \mathbb{Z}^+$, $v_i \in V$, and $f_i(x) \in K[x]$ for $1 \le i \le k$, with*

$$V = \mathrm{Orb}_X(v_1) \oplus \cdots \oplus \mathrm{Orb}_X(v_k),$$

$\mathrm{Min}_X(v_i) = f_i(x)$, and $f_i(x) \mid f_{i+1}$ for $1 \le i < k$.

The polynomials $f_1(x), \ldots, f_k(x)$ are called the *invariant factors* of the matrix X and are unique. The minimal polynomial of X is $f_k(x)$ and the characteristic polynomial of X is the product $f_1(x) \cdots f_k(x)$.

Theorem 2.5. (Hartley and Hawkes (1970), 8.14) *Let K be a field, $V = K^{(n)}$ and fix $X \in K^{(n\times n)}$. Then there exists $k \in \mathbb{Z}^+$, $v_i \in V$, $p_i(x) \in K[x]$, and $e_i \in \mathbb{Z}^+$ for $1 \le i \le k$, with*

$$V = \mathrm{Orb}_X(v_1) \oplus \cdots \oplus \mathrm{Orb}_X(v_k),$$

$\mathrm{Min}_X(v_i) = p_i(x)^{e_i}$, and $p_i(x)$ irreducible for $1 \le i \le k$.

The polynomials $p_1(x)^{e_1}, \ldots, p_k(x)^{e_k}$ are called the *primary invariant factors* of the matrix X and are unique up to a re-ordering.

We now present a basic algorithm used in the rest of the paper. It simultaneously calculates the minimal polynomial of a vector and a basis for the orbit of the vector.

Algorithm MinOrb

Input:
$V = K^{(n)}$, $X \in K^{(n\times n)}$, $v \in V$.

Output:
$\mathrm{Min}_X(v)(x)$ and a basis for $\mathrm{Orb}_X(v)$.

Method:
Simply calculate $v, v \cdot X, v \cdot X^2, \ldots$, forming an echelonized basis of the space spanned

by these vectors, until some linear relation

$$a_0 v + a_1 v \cdot X + \cdots + a_{d-1} v \cdot X^{d-1} + v \cdot X^d = 0$$

is found with d minimal. Then $f(x) = x^d + a_{d-1}x^{d-1} + \cdots + a_0$ is the minimal polynomial of v with respect to X, and $B = \{v, v \cdot X, \ldots, v \cdot X^{d-1}\}$ is a basis for $\mathrm{Orb}_X(v)$.

Proof of Correctness. The algorithm terminates since V has finite dimension. If $\partial \mathrm{Min}_X(v)(x)$ were less than d then that would be detected as a linear relation with fewer vectors than in the result. Thus $\mathrm{Min}_X(v)(x)$ has degree d and since the resulting polynomial is monic of degree d and $\mathrm{Min}_X(v)(x)$ is unique, the resulting polynomial must be $\mathrm{Min}_X(v)(x)$. To see that B is a basis for $\mathrm{Orb}_X(v)$, note that any element of $\mathrm{Orb}_X(v)$ has the form $v \cdot p(X)$ where $p(x) \in K[x]$. By the division algorithm, we may write $p(x) = q(x)\mathrm{Min}_X(v)(x) + r(x)$ with $q(x), r(x) \in K[x]$ and $\partial r(x) < d = \partial \mathrm{Min}_X(v)(x)$. Then $v \cdot p(X) = v \cdot r(X)$ which is certainly in the space spanned by B. If B were not a basis, then there would be a non-trivial dependency amongst the elements of B (which has cardinality d) contradicting the fact that the degree of $\mathrm{Min}_X(v)(x)$ is d. Thus B is a basis for $\mathrm{Orb}_X(v)$. □

The following simple lemmas will find frequent application in the rest of the paper.

Lemma 2.6. *If S and T are X-invariant subspaces of V with $\mathrm{Min}_X(S)(x) = f(x)$, $\mathrm{Min}_X(T)(x) = g(x)$, and $f(x)$ and $g(x)$ are coprime then $S \cap T = \langle 0 \rangle$.*

Proof. Suppose $v \in S \cap T$. Then $v \cdot f(X) = v \cdot g(X) = 0$. Since $f(x)$ and $g(x)$ are coprime, by the Euclidean algorithm there exist $a(x)$ and $b(x)$ in $K[x]$ with $1 = f(x)a(x) + g(x)b(x)$. Then

$$v = v \cdot 1 = v \cdot (f(X)a(X) + g(X)b(X)) = 0.$$

Thus $S \cap T = \langle 0 \rangle$. □

Lemma 2.7. *Suppose $v \in V$, $f(x) \in K[x]$, and $v \cdot f(X) = 0$. Then $\mathrm{Min}_X(v)(x)$ divides $f(x)$.*

Proof. By the division algorithm there exist $q(x), r(x) \in K[x]$ such that $f(x) = q(x)\mathrm{Min}_X(v)(x) + r(x)$, with $0 \leq \partial r(x) < \partial \mathrm{Min}_X(v)(x)$. Then $v \cdot r(X) = v \cdot (f(X) - q(X)\mathrm{Min}_X(v)(X)) = 0$ so $r(x) = 0$ by the minimality of the degree of $\mathrm{Min}_X(v)(x)$. Thus $\mathrm{Min}_X(v)(x)$ divides $f(x)$. □

Lemma 2.8. *For $v \in V$, $\mathrm{Min}_X(\mathrm{Orb}_X(v))(x) = \mathrm{Min}_X(v)(x)$.*

Proof. First note that $v \cdot \mathrm{Min}_X(\mathrm{Orb}_X(v))(X) = 0$ since $v \in \mathrm{Orb}_X(v)$, so

$$\mathrm{Min}_X(v)(x) \mid \mathrm{Min}_X(\mathrm{Orb}_X(v))(x)$$

by Lemma 2.7. Now suppose $w \in \mathrm{Orb}_X(v)$. Then $w = v \cdot g(X)$ for some $g(x) \in K[x]$. Then

$$w \cdot \mathrm{Min}_X(v)(X) = v \cdot g(X)\mathrm{Min}_X(v)(X) = v \cdot \mathrm{Min}_X(v)(X)g(X) = 0$$

so $\partial\mathrm{Min}_X(w)(x) \leq \partial\mathrm{Min}_X(v)(x)$. As w was arbitrary, we must have

$$\partial\mathrm{Min}_X(\mathrm{Orb}_X(v))(x) \leq \partial\mathrm{Min}_X(v)(x).$$

Combining both results gives that $\mathrm{Min}_X(\mathrm{Orb}_X(v))(x) = \mathrm{Min}_X(v)(x)$. □

LEMMA 2.9. *Suppose $v \in V$, $\mathrm{Min}_X(v)(x) = f(x)g(x)$ and $f(x)$ and $g(x)$ are coprime and monic. Then let $w = v \cdot f(X)$ and let $z = v \cdot g(X)$. Then $\mathrm{Min}_X(w)(x) = g(x)$, $\mathrm{Min}_X(z)(x) = f(x)$, $\mathrm{Orb}_X(w) \cap \mathrm{Orb}_X(z) = \langle 0 \rangle$, and $\mathrm{Orb}_X(w) \oplus \mathrm{Orb}_X(z) = \mathrm{Orb}_X(v)$.*

PROOF. $w \cdot g(X) = v \cdot f(X) \cdot g(X) = 0$. So $\mathrm{Min}_X(w)(x)$ divides $g(x)$ by Lemma 2.7. Suppose $\mathrm{Min}_X(w)(x) = h(x)$ with $\partial h(x) < \partial g(x)$. Then $v \cdot f(X)h(X) = 0$ and $\partial f(x)h(x) < f(x)g(x)$, contradicting the minimality of the degree of $f(x)g(x)$. So $\mathrm{Min}_X(w)(x) = g(x)$. A similar argument shows that $\mathrm{Min}_X(z)(x) = f(x)$. By Lemma 2.8, $\mathrm{Min}_X(\mathrm{Orb}_X(w))(x) = g(x)$ and $\mathrm{Min}_X(\mathrm{Orb}_X(v))(x) = f(x)$ so $\mathrm{Orb}_X(w) \cap \mathrm{Orb}_X(v) = \langle 0 \rangle$ by Lemma 2.6. Thus

$$\begin{aligned}\mathrm{Dim}(\mathrm{Orb}_X(w) + \mathrm{Orb}_X(z)) &= \mathrm{Dim}(\mathrm{Orb}_X(w)) + \mathrm{Dim}(\mathrm{Orb}_X(z)) - \\ &\qquad \mathrm{Dim}(\mathrm{Orb}_X(w) \cap \mathrm{Orb}_X(z)) \\ &= \partial g(x) + \partial f(x) - 0 \\ &= \partial(f(x)g(x)) \\ &= \mathrm{Dim}(\mathrm{Orb}_X(v)).\end{aligned}$$

As $\mathrm{Orb}_X(w) + \mathrm{Orb}_X(z) \leq \mathrm{Orb}_X(v)$ and the dimensions of the sum and $\mathrm{Orb}_X(v)$ are the same, equality must hold and since the summands have zero intersection, the sum is direct. □

3. Primary Decomposition

In this section we present the main sub-algorithm used in the canonical form algorithm. It finds a decomposition of the vector space V acted on by X into *primary* subspaces (X-invariant subspaces whose minimal polynomials are powers of irreducible polynomials).

We also "hopefully" obtain cyclic generators for the primary subspaces because of the way the subspaces are constructed. In the process of the algorithm we may have to combine some cyclic subspaces which means that we do not know a cyclic generator for the sum. The next section addresses this problem of not knowing a cyclic generator for a primary subspace.

First we make some observations on the concept of *independent subspaces*.

DEFINITION 3.1. *If $V_1, \ldots, V_k$ are subspaces of a vector space V, then we say that $V_1, \ldots, V_k$ are* independent subspaces *of V if for any $v_1 \in V_1, \ldots, v_k \in V_k$, with all the v_i non-zero, the only solution to*

$$\sum_{i=1}^{k} \lambda_i v_i = 0$$

with $\lambda_i \in K$ is the solution $\lambda_1 = \cdots = \lambda_k = 0$.

Note that it is *not* sufficient for independence that the subspaces are pairwise independent since it is easy to construct subspaces V_1, V_2, and V_3 of V such that $V_1 \cap V_2 =$

$V_1 \cap V_3 = V_2 \cap V_3 = \langle 0 \rangle$ but there exist non-zero $v_1 \in V_1$ and $v_2 \in V_2$ such that $v_1 + v_2 \in V_3$ so that V_1, V_2, and V_3 are not independent.

Note also that if $V_1, \ldots, V_k$ are independent subspaces of a vector space V then and only then does it make sense to write the direct sum

$$V_1 \oplus \cdots \oplus V_k.$$

LEMMA 3.2. *Let $W_1, \ldots, W_k$ be independent subspaces of $V = K^{(n)}$ and let $U_1, \ldots, U_m$ also be independent subspaces of V. Suppose that there exist $f(x), g(x) \in K[x]$ such that $f(x)$ and $g(x)$ are coprime and $\mathrm{Min}_X(W_i)$ divides $f(x)$ for $1 \leq i \leq k$ and $\mathrm{Min}_X(U_i)$ divides $g(x)$ for $1 \leq i \leq m$. Then $W_1, \ldots, W_k, U_1, \ldots, U_m$ are independent subspaces of V.*

PROOF. Let $W = W_1 \oplus \cdots \oplus W_k$ and $U = U_1 \oplus \cdots \oplus U_m$; we may write these direct sums since the summands are independent in each case. It is easy to see that $\mathrm{Min}_X(W)(x)$ divides $f(x)$ and $\mathrm{Min}_X(U)(x)$ divides $g(x)$. Then by Lemma 2.6 we see that $W \cap U = \langle 0 \rangle$. Now suppose there are $w_i \in W_i$ for $1 \leq i \leq k$ and $u_i \in U_i$ for $1 \leq i \leq m$, such that

$$\sum_{i=1}^{k} \lambda_i w_i + \sum_{i=1}^{m} \mu_i u_i = 0$$

with $\lambda_i, \mu_i \in K$ and not all the vectors are zero. Let

$$v = \sum_{i=1}^{k} \lambda_i w_i = -\sum_{i=1}^{m} \mu_i u_i.$$

Then $v \in W$ and $v \in U$ so $v \in W \cap U = \langle 0 \rangle$. Thus $v = 0$. But then $\sum_{i=1}^{k} \lambda_i w_i = 0$ and $\sum_{i=1}^{m} \mu_i u_i = 0$ and by the independence of $W_1, \ldots, W_k$ and the independence of $U_1, \ldots, U_m$ all of the λ_i and μ_i must be zero. Thus $W_1, \ldots, W_k$, $U_1, \ldots, U_m$ are independent. □

We now present the algorithm to find a decomposition of V acted on by X into primary subspaces.

Algorithm DECOMPOSE

Input:
$V = K^{(n)}$, $X \in K^{(n \times n)}$.

Output:
$k \in \mathbb{Z}^+$, $V_i \leq V$, $p_i(x) \in K[x]$, $e_i \in \mathbb{Z}^+$, and $v_i \in V$ for $1 \leq i \leq k$ with V_i X-invariant,

$$V = V_1 \oplus \cdots \oplus V_k,$$

$\mathrm{Min}_X(V_i) = p_i(x)^{e_i}$, $p_i(x)$ irreducible, and $v_i = 0$ or $V_i = \mathrm{Orb}_X(v_i)$ for $1 \leq i \leq k$.

Method:
We shall decompose V into primary subspaces and collect together subspaces whose minimal polynomials are various powers of a *common* irreducible. To do this, we maintain

pairs g_i and L_i $(1 \le i \le r)$ where g_i is an irreducible polynomial from $K[x]$ and L_i is a set of X-invariant subspaces of V whose minimal polynomials are all powers of g_i. Furthermore, the subspaces of V making up the union of all the L_i are always independent.

When a cyclic subspace W of V is constructed we wish to remember the cyclic generator of W. To simplify notation, we introduce an attribute $\mathrm{Gen}(W)$ of W which is a vector in V denoting a cyclic generator of W if it is known, or 0 if it is not known.

Execute the following statements:

$r := 0$;
A: while $(\sum_{i=1}^{r} \sum_{U \in L_i} U) < V$ do
choose $v \in V \setminus (\sum_{i=1}^{r} \sum_{U \in L_i} U)$;
$m(x) := \mathrm{Min}_X(v)(x)$;
$p(x) := 1$;
B: for i := 1 to r do
if $\mathrm{GCD}(m(x), g_i(x)) \neq 1$ then
$f(x) :=$ the maximal power of $g_i(x)$ dividing $m(x)$;
$w := v \cdot (m/f)(X)$;
$p(x) := p(x)f(x)$;
$W := \mathrm{Orb}_X(w)$;
C: compare W with all spaces in L_i:
W is independent with (the sum of) all spaces in L_i:
$\mathrm{Gen}(W) := w$;
$L_i := L_i \cup \{W\}$;
$W \le \sum_{U \in L_i} U$:
/* ignore W; */
W contains some U_j (for $j = 1, \ldots, l$) from L_i but is independent with the rest of the spaces in L_i:
$\mathrm{Gen}(W) := w$;
$L_i := L_i \setminus \{U_j : j \in \{1, \ldots, l\}\}$;
$L_i := L_i \cup \{W\}$;
W has non-trivial intersection with the sum of some U_j (for $j = 1, \ldots, l$) from L_i but is independent with the rest of the spaces in L_i:
$L_i := L_i \setminus \{U_j : j \in \{1, \ldots, l\}\}$;
$T := W + \sum_{j=1}^{l} U_j$;
$\mathrm{Gen}(T) := 0$;
$L_i := L_i \cup \{T\}$;
end compare;
end if;
end for;
D: $v := v \cdot p(X)$;
$m(x) := m(x)/p(x)$;
if $m(x) \neq 1$ then
factorize $m(x)$ as $\prod_{i=1}^{l} q_i(x)^{e_i}$;
E: for i := 1 to l do
$w := v \cdot (m/q_i^{e_i})(X)$;
$W := \mathrm{Orb}_X(w)$;

```
                    Gen(W) := w;
                    r := r + 1;
                    L_r := {W};
                    g_r := q_i;
                end for;
            end if;
        end while;
        k := 0;
F:      for i := 1 to r do
            for U in L_i do
                k := k + 1;
                V_k := U;
                p_k := g_i;
                e_k := multiplicity of g_i(x) in Min_X(U)(x);
                v_k := Gen(V_k);
            end for;
        end for;
```

Proof of Correctness. Note first that the comparison at label C is well defined (at least one of the conditions must be satisfied) since either W is independent with (the sum of) all of the L_i so the first condition is satisfied, or the last condition is trivially satisfied (by noting that W must then have some intersection with the sum of all the L_i). The other conditions are "optimizations" in the sense that the actions taken on each preserve more information if possible; furthermore, the first possible condition satisfied should be taken. If the last condition is taken, the spaces U_j should be as few as possible.

We shall show that the following invariants hold throughout the algorithm.

Invariant (1):
All the spaces from all the L_i are independent.

Invariant (2):
For $1 \leq i \leq r$ and for $U \in L_i$, $\mathrm{Min}_X(U)(x) = g_i(x)^e$ for some e.

First we see that the invariants are trivially established at the initialization of r to 0 (so there are no L_i or g_i yet).

Next we show that the invariants are preserved at label A.

First assume that the invariants hold at the start of the loop beginning at label B. For each i within this loop, we see that $\mathrm{Min}_X(w)(x) = f(x)$ at label C, by applying Lemma 2.9 noting that $m(x)$ and $m(x)/f(x)$ are coprime. Then by Lemma 2.8 we see also that $\mathrm{Min}_X(W)(x) = f(x)$. The comparison step at label C either leaves L_i alone or modifies L_i such that the spaces of L_i are still independent but $W \leq \sum_{U \in L_i} U$ now. Since the minimal polynomials of the spaces from L_i are still coprime with the minimal polynomials of the spaces from all the other L_j, we see by Lemma 3.2 that all the spaces from the L_i are still independent. Thus invariant (1) is preserved by the end of the loop and so at label B. Now L_i is only changed by deleting spaces from it or by inserting into it W or the sum of W with previous spaces of L_i, all of which spaces have minimal

polynomial a power of g_i. Thus invariant (2) is preserved by the end of the loop and so at label `B`.

After the two assignments at label `D`, it is easy to see that $\mathrm{Min}_X(v)(x) = m(x)$ still by applying Lemma 2.9 noting that $p(x)$ and $m(x)/p(x)$ are coprime since $p(x)$ consists of the product of all the maximal powers of the irreducible polynomials g_i occurring in $m(x)$. After the assignments, $m(x)$ is now coprime with all of the $g_i(x)$.

Now assume that the invariants hold at the start of the loop `E`. For each i we see that again $\mathrm{Min}_X(w)(x) = q_i(x)^{e_i}$ by applying Lemma 2.9 since $q_i(x)^{e_i}$ and $m(x)/q_i(x)^{e_i}$ are coprime. Then by Lemma 2.8 we see that $\mathrm{Min}_X(W)(x) = q_i(x)^{e_i}$ also. So incrementing r and then setting L_r to be $\{W\}$ and $g_r = q_i$ preserves invariant (1) since the minimal polynomials of the spaces in the other L_j are coprime with the minimal polynomial of W so we can again apply Lemma 3.2. Invariant (2) is clearly preserved since $\mathrm{Min}_X(W)(x)$ is a power of g_r.

Since the invariants are preserved at labels `B` and `D`, it follows that they are preserved at label `A`.

We now show that the algorithm terminates. The while loop at label `A` continues as long as the (direct) sum of all the spaces from the L_i is strictly less than V. In such a case, a vector v is chosen from V such that v is not in that sum. It is easy to see that after the body of the loop is executed, the whole of $\mathrm{Orb}_X(v)$ is included in the sum of all the spaces in the L_i since in Lemma 2.9 the orbit of the original vector equals the direct sum of the orbits of the new vectors and that Lemma is applied effectively to split $\mathrm{Orb}_X(v)$ into primary spaces which are then combined with the previous spaces in the L_i or are inserted into some L_i. Thus the sum of all the spaces from the L_i properly increases each time so the loop at label `A` must terminate.

Finally, since the loop at label `A` terminates with the (direct) sum of all the spaces from the L_i equal to V and the invariants hold at the end of that loop, it is clear that the assignments of V_k, p_k, e_k, and v_k within the loop beginning at label `F` satisfy the conditions asserted for the output. □

4. Splitting a Primary Space

In this section we attack the problem which arose in the last section—we may have an X-invariant subspace of our vector space V which is primary under the action of X but we do not know a cyclic generator for it. (Indeed, there may not exist one—the subspace may have to be split further to obtain a cyclic decomposition.) The algorithm in this section finds a complete cyclic decomposition of an X-invariant subspace under the action of a matrix X, assuming that the action is *primary*.

Algorithm SplitPrimary

Input:
$W \leq K^{(n)}$, $X \in K^{(n\times n)}$, $f(x) \in K[x]$, $q \in \mathbb{Z}^+$, with W X-invariant, $\mathrm{Min}_X(W)(x) = f(x)^q$, $f(x)$ irreducible.

Output:
$m_1, \ldots, m_q \in \mathbb{Z}^+$, $w_{i,j} \in W$ for $1 \leq i \leq q$ and $1 \leq j \leq m_i$ such that

$$W = \mathrm{Orb}_X(w_{1,1}) \oplus \cdots \oplus \mathrm{Orb}_X(w_{1,m_1}) \oplus \cdots \oplus \mathrm{Orb}_X(w_{k,1}) \oplus \cdots \oplus \mathrm{Orb}_X(w_{k,m_k})$$

and $\mathrm{Min}_X(w_{i,j}) = f(x)^i$ for $1 \leq i \leq q$ and $1 \leq j \leq m_i$.

Method:
Let $N = f(X)$, and $W_i = \ker(N^i) \cap W$ for $0 \leq i \leq q$. (N is said to be nilpotent of index q in its action on W.) Thus

$$\langle 0 \rangle = W_0 \leq W_1 \leq \ldots \leq W_q = W.$$

Then for $1 \leq i \leq q$, let C_i be a complement of W_{i-1} in W_i. Thus

$$W_i = C_1 \oplus C_2 \oplus \cdots \oplus C_i.$$

Let $d = \partial f(x)$ and let B_i be a basis for C_i. Then execute the following statements:

```
    S_{q+1} := {};
A:  for i := q to 1 by −1 do
B:      S_i := S_{i+1} · N;
        m_i := 0;
        for w in B_i do
            if w ∉ ⟨S_i⟩ ⊕ W_{i−1} then
C:              S_i := S_i ∪ {w, ..., w · X^{d−1}};
                m_i := m_i + 1;
                w_{i,m_i} := w;
            end if;
        end for;
    end for;
```

It is quite difficult to show that the algorithm produces a decomposition of W into a direct sum of cyclic subspaces. We first need some technical lemmas and then we present some theorems which establish the correctness of the algorithm.

First note that for each i, W_i is an X-invariant (and thus also N-invariant) subspace of W since $w \in W_i$ implies that $w \cdot N^i = 0$ so $(w \cdot X)N^i = w \cdot (XN^i) = 0$ giving that $w \cdot X \in W_i$ also. We also define

$$[w]_d = \{w, \ldots, w \cdot X^{d-1}\}$$

and

$$\langle w \rangle_d = \langle w, \ldots, w \cdot X^{d-1} \rangle$$

since these will be used often in the following.

LEMMA 4.1. *Let $1 \leq i \leq q$ and suppose $w \in W_i \setminus W_{i-1}$. Then $\mathrm{Min}_X(w)(x) = f(x)^i$.*

PROOF. $w \in W_i \setminus W_{i-1} = \ker(N^i) \setminus \ker(N^{i-1})$ so $\mathrm{Min}_N(w) = x^i$. Thus $w \cdot N^i = 0$ with i minimal. As $N = f(X)$, $w \cdot f(X)^i = 0$ so $\mathrm{Min}_X(w)(x) \mid f(x)^i$ by Lemma 2.7 or $\mathrm{Min}_X(w)(x) = f(x)^j$ with $0 \leq j \leq i$ since $f(x)$ is irreducible. Then $w \cdot f(X)^j = 0$ with $0 \leq j \leq i$ so $w \cdot N^j = 0$ which implies that $j = i$ since i is minimal. Thus $\mathrm{Min}_X(w)(x) = f(x)^i$. □

LEMMA 4.2. *Let $1 \leq i \leq q$ and suppose $w \in W_i \setminus W_{i-1}$. Then $\langle w \rangle_d \setminus \{0\} \subseteq W_i \setminus W_{i-1}$.*

PROOF. Suppose $u \in \langle w \rangle_d$ and $u \neq 0$. Then $u = w \cdot p(X)$ for some $p(x) \in K[x]$ with $p(x) \neq 0$ and $\partial p(x) < d$ by the definition of $\langle w \rangle_d$. Now $u \in W_j \setminus W_{j-1}$ for a unique j $(1 \leq j \leq q)$ so $u \cdot N^j = 0$, and $u \cdot N^{j-1} \neq 0$. So $u \cdot f(X)^j = 0$ or $w \cdot p(X) \cdot f(X)^j = 0$. Then if $j < i$, then by Lemma 2.7 $\partial \mathrm{Min}_X(w)(x) \leq \partial p(x) + jd < id$ (since $\partial p(x) < d$ and $j < i$) which contradicts Lemma 4.1. So $j = i$ and $u \in W_i \setminus W_{i-1}$. □

LEMMA 4.3. *Let $1 \leq i \leq q$ and suppose $w \in W_i \setminus W_{i-1}$. Suppose $u \in \langle w \rangle_d$ with $u \neq 0$. Then there exists $p(x) \in K[x]$ and $t \in W_{i-1}$ such that $w = u \cdot p(X) + t$.*

PROOF. Since $u \in \langle w \rangle_d$ and $u \neq 0$, there exists $g(x) \in K[x]$ with $g(x) \neq 0$ and $\partial g(x) < \partial f(x)$ such that $u = w \cdot g(X)$. Since $f(x)$ is irreducible, $f(x)$ and $g(x)$ are coprime, so by the Euclidean algorithm there exist polynomials $p(x), q(x) \in K[x]$ with $p(x)g(x) + q(x)f(x) = 1$. Then

$$u \cdot p(X) = w \cdot g(X)p(X) = w - w \cdot f(X)q(X).$$

Thus if we take $t = -w \cdot f(X)q(X)$ then $w = u \cdot p(X) + t$ as desired (t is clearly in W_{i-1} since $w \cdot f(X) = w \cdot N \in W_{i-1}$). □

LEMMA 4.4. *Let $1 \leq i \leq q$ and suppose $w \in W_i \setminus W_{i-1}$. Suppose $u \in \langle w \rangle_d$. Then for any $g(x) \in K[x]$, $u \cdot g(X) \in \langle w \rangle_d \oplus W_{i-1}$.*

PROOF. (First note that $\langle w \rangle_d \setminus \{0\} \subseteq W_i \setminus W_{i-1}$ by Lemma 4.2 so $\langle w \rangle_d \cap W_{i-1} = \langle 0 \rangle$ so the sum of $\langle w \rangle_d$ and W_{i-1} is direct.) Since $u \in \langle w \rangle_d$, there exists $h(x) \in K[x]$ such that $u = w \cdot h(X)$. By the division algorithm, we can write $g(x)h(x) = f(x)q(x) + r(x)$ for some $q(x), r(x) \in K[x]$ with $\partial r(x) < \partial f(x)$. Then

$$\begin{aligned} u \cdot g(X) &= w \cdot h(X)g(X) \\ &= w \cdot f(X)q(X) + w \cdot r(X). \end{aligned}$$

Now $w \cdot f(X) = w \cdot N \in W_{i-1}$ since $w \in W_i$ so $w \cdot f(X) \cdot q(X) \in W_{i-1}$ also (W_{i-1} is an X-invariant subspace) and $w \cdot r(X) \in \langle w \rangle_d$ since $\partial r(x) < \partial f(x) = d$. Thus $u \cdot g(X) \in \langle w \rangle_d \oplus W_{i-1}$. □

We can now prove the critical theorem about the sets S_i which are constructed by the algorithm.

THEOREM 4.5. *At the conclusion of the algorithm, S_i is a basis for a complement to W_{i-1} in W_i for $1 \leq i \leq q$.*

PROOF. Clearly S_i is formed during step i of the loop beginning at label `A`. We shall show that the following invariants hold within step i of the loop.

Invariant (1):
$\langle S_i \rangle \setminus \{0\} \subseteq W_i \setminus W_{i-1}$.

Invariant (2):
$\langle S_i \rangle = \sum_{j=1}^{r_i} \langle z_{i,j} \rangle_d$ for some r_i and some $z_{i,j} \in W_i \setminus W_{i-1}$.

Invariant (3):
S_i is a basis (i.e. the elements of S_i are independent).

To show that the invariants hold, we must first show that the initialization of S_i at label `B` establishes each of the invariants. Then we must show that the assignment statement at label `C` preserves each of the invariants (assuming they hold before the statement). Also, for $i < q$, we may assume by induction that all of the invariants hold for S_{i+1} since S_{i+1} is constructed before S_i.

We first show that the invariants for S_i are established at the initialization of S_i at label `B`. For $i = q$, S_i is initialized to $\{\}$ which trivially satisfies all of the invariants (take $r_i = 0$ for invariant (2)). So suppose $i < q$. As invariant (1) holds for $i+1$ by induction, $\langle S_{i+1}\rangle \backslash \{0\} \subseteq W_{i+1} \backslash W_i$. At label `B`, S_i is initialized to $S_{i+1} \cdot N$. Suppose $w \in \langle S_{i+1}N\rangle \backslash \{0\}$. Then $w = u \cdot N$ for some $u \in \langle S_{i+1}\rangle \setminus \{0\} \subseteq W_{i+1} \setminus W_i$. So $u \in \ker(N^{i+1})$ or $u \cdot N^{i+1} = 0$ but $u \cdot N^i \neq 0$. So $w \cdot N^i = u \cdot N^{i+1} = 0$ but $w \cdot N^{i-1} = u \cdot N^i \neq 0$. Thus $w \in W_i \setminus W_{i-1}$. Thus after the assignment at label `B`, invariant (1) is established.

Now define a map $\phi : \langle S_{i+1}\rangle \to W_i$ by $\phi(w) = w \cdot N$. By the discussion in the last paragraph, ϕ is well defined, that is, the image of an element of $\langle S_{i+1}\rangle$ under ϕ lies in W_i. Clearly ϕ is a vector space homomorphism. Now suppose $w, u \in \langle S_{i+1}\rangle$ with $w \neq u$ and $\phi(w) = \phi(u)$. Then $(w - u) \cdot N = 0$ so $w - u \in \ker(N) = W_1$ with $w - u \neq 0$. But $w - u \in \langle S_{i+1}\rangle \setminus \{0\} \subseteq W_{i+1} \setminus W_i$ which excludes W_1 since $i \geq 1$. This gives a contradiction. Thus ϕ is a monomorphism.

Since by induction invariant (2) holds for $i+1$, we have that $\langle S_{i+1}\rangle = \sum_{j=1}^{r_{i+1}} \langle z_{i+1,j}\rangle_d$ for some r_{i+1} and some $z_{i+1,j} \in W_{i+1} \setminus W_i$. Then by initializing r_i to r_{i+1} and $z_{i,j}$ to $z_{i+1,j} \cdot N$ for $j = 1, \ldots, r_i$ and using the fact that ϕ is a monomorphism, we have that

$$\langle S_i\rangle = \langle S_{i+1} \cdot N\rangle = \langle \phi(S_{i+1})\rangle = \phi(\langle S_{i+1}\rangle),$$

which, by our induction hypothesis, equals

$$\phi\left(\sum_{j=1}^{r_{i+1}} \langle z_{i+1,j}\rangle_d\right) = \sum_{j=1}^{r_{i+1}} \phi(\langle z_{i+1,j}\rangle_d) = \sum_{j=1}^{r_i} \langle z_{i,j}\rangle_d.$$

Thus after the assignment at label `B`, invariant (2) is established.

Since S_{i+1} is a basis by induction, and a monomorphism maps one set of independent vectors to another set of independent vectors, after the assignment at label `B` S_i must be initialized to a basis so invariant (3) is established. Thus each of the invariants is established after the initialization assignment at label `B`.

We now proceed to show that at label `C` the invariants are preserved. We can assume that they hold for S_i and S_{i+1} before the assignment by induction. Now the assignment at label `C` inserts $[w]_d$ in S_i where w is chosen from C_i such that $w \notin \langle S_i\rangle \oplus W_{i-1}$. Note that since invariant (1) holds by assumption, $\langle S_i\rangle$ and W_{i-1} have zero intersection so the sum of $\langle S_i\rangle$ and W_{i-1} is direct.

As $w \in C_i \setminus \{0\} \subseteq W_i \setminus W_{i-1}$, Lemma 4.2 implies that $\langle w\rangle_d \setminus \{0\} \subseteq W_i \setminus W_{i-1}$. Thus after inserting $[w]_d$ in S_i at label `C`, invariant (1) is preserved.

By incrementing r_i and then setting z_{i,r_i} to be w after $[w]_d$ is inserted in S_i, we clearly see that invariant (2) is also preserved at label `C`.

As $w \in C_i \setminus \{0\} \subseteq W_i \setminus W_{i-1}$, $\mathrm{Min}_X(w)(x) = f(x)^i$ by Lemma 4.1 and $i \geq 1$, so $w, \ldots, w \cdot X^{d-1}$ are independent. We now show that $\langle w\rangle_d \cap (\langle S_i\rangle \oplus W_{i-1}) = \langle 0\rangle$. Now $w \notin \langle S_i\rangle \oplus W_{i-1}$ by choice. Suppose $u \in \langle w\rangle_d \cap (\langle S_i\rangle \oplus W_{i-1})$ and $u \neq 0$. Then $u \in \langle w\rangle_d$

so by Lemma 4.3 there exists $p(x) \in K[x]$ and $t \in W_{i-1}$ such that $w = u.p(X) + t$. Now $u \in \langle S_i \rangle \oplus W_{i-1}$ also so by using invariant (2), we may write

$$u = \left(\sum_{j=1}^{r_i} y_{i,j} \right) + s$$

with $y_{i,j} \in \langle z_{i,j} \rangle_d$ and $s \in W_{i-1}$. Thus

$$\begin{aligned} w &= u \cdot p(X) + t \\ &= \left(\sum_{j=1}^{r_i} y_{i,j} \right) \cdot p(X) + s \cdot p(X) + t \\ &= \left(\sum_{j=1}^{r_i} y_{i,j} \cdot p(X) \right) + s \cdot p(X) + t. \end{aligned}$$

Now since $y_{i,j} \in \langle z_{i,j} \rangle_d$ and $z_{i,j} \in W_i \setminus W_{i-1}$, we may apply Lemma 4.4 to see that $y_{i,j} \cdot p(X) \in \langle z_{i,j} \rangle_d \oplus W_{i-1}$. Thus every term of the sum lies in $\langle S_i \rangle \oplus W_{i-1}$ and $s \cdot p(X) + t$ clearly lies in W_{i-1}. Thus $w \in \langle S_i \rangle \oplus W_{i-1}$ which is a contradiction. Thus $\langle w \rangle_d \cap (\langle S_i \rangle \oplus W_{i-1}) = \langle 0 \rangle$ so after inserting $[w]_d$ in S_i at label `C`, the vectors of S_i are still independent so invariant (3) is preserved.

Thus we see that at the conclusion of the algorithm all of the invariants hold for each i.

Fix i such that $1 \leq i \leq q$. Then $\langle S_i \rangle \subseteq W_i$ and $\langle S_i \rangle \cap W_{i-1} = \langle 0 \rangle$ by invariant (1). So

$$\langle S_i \rangle \oplus W_{i-1} \subseteq W_i.$$

Also, at the conclusion of the algorithm we must have that $C_i \subseteq \langle S_i \rangle \oplus W_{i-1}$ since any basis element of C_i not already in $\langle S_i \rangle \oplus W_{i-1}$ is inserted in S_i at label `C`. Thus

$$W_i = C_i \oplus W_{i-1} \subseteq (\langle S_i \rangle \oplus W_{i-1}) + W_{i-1} = \langle S_i \rangle \oplus W_{i-1}.$$

Thus $\langle S_i \rangle \oplus W_{i-1} = W_i$ and since by invariant (3) S_i is a basis, we finally have that at the conclusion of the algorithm, S_i is a basis for a complement to W_{i-1} in W_i. □

We immediately obtain:

Corollary 4.6. *Write*

$$S = S_1 \dot{\cup} S_2 \dot{\cup} \cdots \dot{\cup} S_q.$$

Then at the conclusion of the algorithm, S is a basis for W.

We now characterize the sets S_i in terms of the $w_{j,l}$ vectors which form the output of the algorithm.

Lemma 4.7. *At the conclusion of the algorithm,*

$$S_i = \bigcup_{j=i}^{q} \bigcup_{l=1}^{m_j} [w_{j,l} \cdot N^{j-i}]_d$$

for $1 \leq i \leq q$.

PROOF. First note that by defining S_{q+1} to be $\{\}$, the statement also holds for S_{q+1}. We can then prove the statement by induction on i from q down to 1 assuming the statement is true for $i+1$. S_i is initialized at label `B` to $S_{i+1} \cdot N$ and thereafter, at label `C`, exactly $[w_{i,1}]_d, [w_{i,2}]_d, \ldots,$ and $[w_{i,m_i}]_d$ are inserted in S_i. Thus,

$$\begin{aligned} S_i &= S_{i+1} \cdot N \cup \left(\bigcup_{l=1}^{m_i} [w_{i,l}]_d \right) \\ &= \left(\bigcup_{j=i+1}^{q} \bigcup_{l=1}^{m_j} [w_{j,l}.N^{j-(i+1)}]_d \right) \cdot N \cup \left(\bigcup_{l=1}^{m_i} [w_{i,l}]_d \right) \qquad \text{(by induction)} \\ &= \left(\bigcup_{j=i+1}^{q} \bigcup_{l=1}^{m_j} [w_{j,l} \cdot N^{j-i}]_d \right) \cup \left(\bigcup_{l=1}^{m_i} [w_{i,l}]_d \right) \\ &= \bigcup_{j=i}^{q} \bigcup_{l=1}^{m_j} [w_{j,l} \cdot N^{j-i}]_d. \end{aligned}$$

Thus the statement holds for all i from q down to 1. □

LEMMA 4.8. *Let $1 \leq i \leq q$ and suppose $w \in W_i \setminus W_{i-1}$. Let B be*

$$[w]_d \cup [w \cdot N]_d \cup \cdots \cup [w \cdot N^{i-1}]_d.$$

Then B is a basis for $\mathrm{Orb}_X(w)$.

PROOF. By Lemma 4.1, $\mathrm{Min}_X(w)(x) = f(x)^i$. B consists of the vectors $w \cdot N^j X^l$ for $0 \leq j < i$ and $0 \leq l < d$. If B were not a basis, then a dependency amongst the elements of B would yield the equation

$$\sum_{j=0}^{i-1} \sum_{l=0}^{d-1} \alpha_{j,l} w \cdot N^j X^l = 0$$

with the $\alpha_{j,l} \in K$ not all zero, and since $N = f(X)$, the equation could be written as $w \cdot p(X) = 0$ where $p(x) \in K[x]$, $p(x) \neq 0$, and $\partial p(x) < id$, contradicting the fact that $\partial \mathrm{Min}_X(w)(x) = \partial(f(x)^i) = id$. Thus B is a basis, and the dimension of $\langle B \rangle$ is id. But clearly $\langle B \rangle \leq \mathrm{Orb}_X(w)$ and the dimension of $\mathrm{Orb}_X(w)$ is id (since $\mathrm{Min}_X(w)(x) = f(x)^i$), so $\langle B \rangle = \mathrm{Orb}_X(w)$. Thus B is a basis for $\mathrm{Orb}_X(w)$. □

Finally, we can prove the correctness of the algorithm.

THEOREM 4.9. *At the conclusion of the algorithm,*

$$W = \mathrm{Orb}_X(w_{1,1}) \oplus \cdots \oplus \mathrm{Orb}_X(w_{1,m_1}) \oplus \cdots \oplus \mathrm{Orb}_X(w_{k,1}) \oplus \cdots \oplus \mathrm{Orb}_X(w_{k,m_k}),$$

and $\mathrm{Min}_X(w_{i,j}) = f(x)^i$ for $1 \leq i \leq q$ and $1 \leq j \leq m_i$ so the output of the algorithm is correct.

PROOF. Consider the following diagram D consisting of vectors from W:

q	$[w_{q,1}]_d$	$[w_{q,1} \cdot N]_d$	$\dots$	$[w_{q,1} \cdot N^{q-1}]_d$
	$\vdots$	$\vdots$	$\vdots$	$\vdots$
	$[w_{q,m_q}]_d$	$[w_{q,m_q} \cdot N]_d$	$\dots$	$[w_{q,m_q} \cdot N^{q-1}]_d$
$q-1$		$[w_{q-1,1}]_d$	$\dots$	$[w_{q-1,1} \cdot N^{q-2}]_d$
		$\vdots$	$\vdots$	$\vdots$
		$[w_{q-1,m_{q-1}}]_d$	$\dots$	$[w_{q-1,m_{q-1}} \cdot N^{q-2}]_d$
			$\ddots$	$\vdots$
1				$[w_{1,1}]_d$
				$\vdots$
				$[w_{1,m_1}]_d$
	S_q	S_{q-1}		S_1

(An empty block indicates that it contains no vectors.) By Lemma 4.7, the column of blocks labelled S_i consists of exactly the vectors in S_i for each i such that $q \geq i \geq 1$. Thus by Corollary 4.6, the vectors making up D form a basis of W.

On the other hand, the rows of D are precisely

$$[w_{i,j}]_d \cup [w_{i,j} \cdot N]_d \cup \cdots \cup [w_{i,j} \cdot N^{i-1}]_d$$

where $q \geq i \geq 1$ and $1 \leq j \leq m_i$. Thus by Lemma 4.8 the rows of D are exactly bases for the orbits $\mathrm{Orb}_X(w_{i,j})$ where $q \geq i \geq 1$ and $1 \leq j \leq m_i$. Thus

$$\mathrm{Orb}_X(w_{1,1}) \oplus \cdots \oplus \mathrm{Orb}_X(w_{1,m_1}) \oplus \cdots \oplus \mathrm{Orb}_X(w_{k,1}) \oplus \cdots \oplus \mathrm{Orb}_X(w_{k,m_k}) = W.$$

The assertion about the minimal polynomials of the $w_{i,j}$ follows from Lemma 4.1. □

5. Combining Spaces

Obtaining a decomposition of our vector space V acted on by X into the direct sum of cyclic primary spaces is precisely what we need to construct the Primary Rational and Jordan forms. However, to construct the Rational form we need also to combine the spaces in this direct sum so that the minimal polynomials of the spaces satisfy the divisibility condition on the invariant factors mentioned in Theorem 2.4.

First we show how to combine spaces having coprime minimal polynomials.

LEMMA 5.1. *Suppose* $v, w \in V$, $\mathrm{Min}_X(v)(x) = f(x)$, $\mathrm{Min}_X(w)(x) = g(x)$, *and* $f(x)$ *and* $g(x)$ *are coprime. Let* $u = v + w$. *Then* $\mathrm{Min}_X(u)(x) = f(x)g(x)$ *and* $\mathrm{Orb}_X(u) = \mathrm{Orb}_X(v) \oplus \mathrm{Orb}_X(w)$.

PROOF. First note that the sum of $\mathrm{Orb}_X(v)$ and $\mathrm{Orb}_X(w)$ is direct by Lemmas 2.8 and 2.6.

Since $f(x)$ and $g(x)$ are coprime, by the Euclidean algorithm there exist $a(x)$ and $b(x) \in K[x]$ with $1 = a(x)f(x) + b(x)g(x)$. Then

$$\begin{aligned} u \cdot a(X)f(X) &= v \cdot a(X)f(X) + w \cdot a(X)f(X) \\ &= w \cdot a(X).f(X) \\ &= w \cdot (1 - b(X)g(X)) \\ &= w, \end{aligned}$$

so $w \in \mathrm{Orb}_X(u)$, and similarly $u \cdot b(X)g(X) = v$, so $v \in \mathrm{Orb}_X(u)$. Thus we have $w \cdot \mathrm{Min}_X(\mathrm{Orb}_X(u))(X) = 0$ and $v \cdot \mathrm{Min}_X(\mathrm{Orb}_X(u))(X) = 0$. Since $\mathrm{Min}_X(w)(x) = g(x)$, by Lemma 2.7 $g(x)$ divides $\mathrm{Min}_X(\mathrm{Orb}_X(u))(x)$, and similarly $f(x)$ divides $\mathrm{Min}_X(\mathrm{Orb}_X(u))(x)$. Since $f(x)$ and $g(x)$ are coprime, their product $f(x)g(x)$ also must divide $\mathrm{Min}_X(\mathrm{Orb}_X(u))(x)$, which equals $\mathrm{Min}_X(u)(x)$ by Lemma 2.8.

On the other hand, clearly $u \cdot f(X)g(X) = 0$ so $\mathrm{Min}_X(u)(x) \mid f(x)g(x)$ by Lemma 2.7. Thus $\mathrm{Min}_X(u)(x) = f(x)g(x)$.

Finally, $\mathrm{Orb}_X(v) \oplus \mathrm{Orb}_X(w) \leq \mathrm{Orb}_X(u)$ since $v = u \cdot f(X) \in \mathrm{Orb}_X(u)$ and $w = u \cdot g(X) \in \mathrm{Orb}_X(u)$. But

$$\begin{aligned}\mathrm{Dim}(\mathrm{Orb}_X(u)) &= \partial \mathrm{Min}_X(u)(x) \\ &= \partial(f(x)g(x)) \\ &= \partial f(x) + \partial g(x) \\ &= \mathrm{Dim}(\mathrm{Orb}_X(v)) + \mathrm{Dim}(\mathrm{Orb}_X(w)).\end{aligned}$$

So $\mathrm{Orb}_X(v) \oplus \mathrm{Orb}_X(w) = \mathrm{Orb}_X(u)$. □

We now present an algorithm which will combine all the cyclic primary spaces into spaces which satisfy the desired properties for the construction of the Rational form.

Algorithm CombineSpaces

Input:
$V = K^{(n)}$, $X \in K^{(n\times n)}$, $k \in \mathbb{Z}^+$, $m_1, \ldots, m_k \in \mathbb{Z}^+$, $v_{i,j} \in V$ and $e_{i,j} \in \mathbb{Z}^+$ for $1 \leq i \leq k$ and $1 \leq j \leq m_i$, $p_i(x) \in K[x]$ for $1 \leq i \leq k$, such that

$$V = \mathrm{Orb}_X(v_{1,1}) \oplus \cdots \oplus \mathrm{Orb}_X(v_{1,m_1}) \oplus \cdots \oplus \mathrm{Orb}_X(v_{k,1}) \oplus \cdots \oplus \mathrm{Orb}_X(v_{k,m_k}),$$

$\mathrm{Min}_X(v_{i,j}) = p_i(x)^{e_{i,j}}$ for $1 \leq i \leq k$ and $1 \leq j \leq m_i$, $p_i(x)$ is irreducible for $1 \leq i \leq k$, and $e_{i,j} < e_{i,j+1}$ for $1 \leq i \leq k$ and $1 \leq j < m_i$.

Output:
$r \in \mathbb{Z}^+$, $w_i \in V$ and $f_i(x) \in K[x]$ for $1 \leq i \leq r$ such that

$$V = \mathrm{Orb}_X(w_1) \oplus \cdots \oplus \mathrm{Orb}_X(w_r),$$

$\mathrm{Min}_X(w_i) = f_i(x)$ for $1 \leq i \leq r$, and

$$f_1(x) \mid f_2(x) \mid \cdots \mid f_r(x).$$

Method:

```
    r := max_{i=1}^{k} m_q;
A:  for l := r to 1 by −1 do
        w := 0;
        f(x) := 1;
B:      for i := 1 to k do
            if m_i > 0 then
C:              w := w + w_{i,m_i};
                f(x) := f(x) · p_i^{e_{i,m_i}}(x);
```

$m_i := m_i - 1$;
end if;
end for;
$w_l := w$;
$f_l(x) := f(x)$;
end for;

Proof of Correctness. Informally, we are given k towers and tower i ($1 \leq i \leq k$) contains m_i blocks and each block contains a vector from V such that the minimal polynomial of that vector is a power of the irreducible polynomial corresponding to that block; furthermore, the powers ascend as one goes up the blocks of a specific tower. The algorithm repeatedly takes a block from the top of each tower (if the tower is non-empty) and combines these blocks into one block. The algorithm finishes when all the towers are exhausted.

We first show that throughout the loop starting at label `A`, $\mathrm{Min}_X(w)(x) = f(x)$. This is obviously established at the initialization of w and $f(x)$. Within the loop starting at label `B`, $f(x)$ and $p_i(x)^{e_{i,m_i}}$ must always be coprime since $f(x)$ is a product of powers of $p_j(x)$ for $1 \leq j < i$ and the $p_j(x)$ are coprime with $p_i(x)$. So Lemma 5.1 is applicable and the new values of w and $f(x)$ still satisfy the invariant. Thus at the end of the loop starting at label `A`, $\mathrm{Min}_X(w_l) = f_l(x)$.

It is clear that by taking r to be the maximum of the m_i for $1 \leq i \leq k$ then the orbits under X of the r vectors w_l ($1 \leq l \leq r$) at the conclusion add up to V as a direct sum since the orbits of the $v_{i,j}$ add up to V as a direct sum and each $v_{i,j}$ is considered at label `C`.

Finally, it is easy to see that $f_l \mid f_{l+1}$ for $1 \leq l < r$ since the multiplicity of a factor $p_i(x)$ in f_l must be less than or equal to the corresponding multiplicity in f_{l+1} since the $e_{i,j}$ are increasing for a fixed i. □

6. Canonical Forms

Finally in this section we can describe each of the canonical forms and present the main algorithm to compute them.

Suppose $f(x) \in K[x]$ such that $f(x)$ is monic and non-zero. Let $d = \partial f(x)$ and write

$$f(x) = x^d + \sum_{i=0}^{d-1} c_i x^i$$

with $c_i \in K$. We define the *companion matrix* $\mathrm{C}(f(x))$ of $f(x)$ to be the $d \times d$ matrix

$$\begin{pmatrix} 0 & 1 & 0 & \dots & 0 \\ 0 & 0 & 1 & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ -c_0 & -c_1 & -c_2 & \dots & -c_{d-1} \end{pmatrix}.$$

It is easy to show that the minimal (and characteristic) polynomial of $\mathrm{C}(f(x))$ is $f(x)$.

In order to describe the (generalized) Jordan form, we must first define some auxiliary

matrices. For $d \geq 1$, let N_d be the $d \times d$ matrix

$$\begin{pmatrix} 0 & 0 & \dots & 0 \\ 0 & 0 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 1 & 0 & \dots & 0 \end{pmatrix},$$

that is, a $d \times d$ matrix having zeros everywhere except for a one in the bottom left corner. Then for a monic non-zero polynomial $p(x) \in K[x]$ and a positive integer $e \geq 1$, we define the *Jordan block* $\mathrm{J}(p(x), e)$ to be the $e \cdot \partial p(x) \times e \cdot \partial p(x)$ matrix

$$\begin{pmatrix} \mathrm{C}(p(x)) & \mathrm{N}_{\partial p(x)} & \dots & 0 & 0 \\ 0 & \mathrm{C}(p(x)) & \dots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \dots & \mathrm{C}(p(x)) & \mathrm{N}_{\partial p(x)} \\ 0 & 0 & \dots & 0 & \mathrm{C}(p(x)) \end{pmatrix},$$

where there are e companion matrix blocks on the diagonal. (Notice that if $p(x)$ is the linear polynomial $x - \lambda$, then the Jordan block $\mathrm{J}(p(x), e)$ has λ on the diagonal, ones above the diagonal and zeros elsewhere which is the common definition of the Jordan block when describing the true Jordan Form over an algebraically closed field.)

We can now describe the canonical forms computed by the algorithm in this paper. A matrix from $K^{(n \times n)}$ is a *Rational* form if it consists of a diagonal block joining of companion matrices such that for each block before the last block, the polynomial corresponding to that block divides the polynomial corresponding to the next block, just like the property of the invariant factors mentioned in Theorem 2.4.

A matrix from $K^{(n \times n)}$ is a *Primary Rational* form if it consists of a diagonal block joining of companion matrices such that the polynomial corresponding to each block is a power of an irreducible polynomial, just like the property of the primary invariant factors mentioned in Theorem 2.5.

A matrix from $K^{(n \times n)}$ is a *generalized Jordan* form if it consists of a diagonal block joining of Jordan blocks (each of which is derived from a power of an irreducible polynomial). If the field K is algebraically closed, so all irreducible polynomials over K are linear, then the generalized Jordan form is the same as the usual definition of the Jordan form.

By appropriate applications of Theorems 2.4 and 2.5, any matrix from $K^{(n \times n)}$ is similar to a matrix which is of each of these forms. The Rational form tends to have the companion matrix blocks as large as possible, while the Primary Rational form tends to have the companion matrix blocks as small as possible. The Jordan form goes just a little further than the Primary Rational form in making the blocks smaller at the price of having the extra ones above the diagonal between the companion matrices in each Jordan block. The Rational form is unique because of the divisibility condition. If we impose any total ordering on $K[x]$ (which need not respect any algebraic structure of $K[x]$) and sort the primary invariant factors of our matrix with respect to this ordering and then the multiplicity of the powers, we can ensure that the Primary Rational and Jordan forms are unique. In the implementation of the algorithm in Magma, total orders on the polynomial rings $K[x]$ for each possible field K have been easily designed.

In order also to describe the transformation matrices produced in the main algorithm,

we define

$$\mathrm{TR}_X(v,d) = \begin{pmatrix} v \\ v\cdot X \\ \vdots \\ v\cdot X^{d-1} \end{pmatrix}, \qquad \text{and} \qquad \mathrm{TJ}_X(v,p(x),e) = \begin{pmatrix} v \\ v\cdot X \\ \vdots \\ v\cdot X^{\partial p-1} \\ v\cdot p(X) \\ v\cdot p(X)X \\ \vdots \\ v\cdot p(X)X^{\partial p-1} \\ \vdots \\ v\cdot p(X)^{e-1} \\ v\cdot p(X)^{e-1}\cdot X \\ \vdots \\ v\cdot p(X)^{e-1}X^{\partial p-1} \end{pmatrix}.$$

Finally, we present the main algorithm to compute canonical forms.

Algorithm CanonicalForm

Input:
$V = K^{(n)}$, $X \in K^{(n\times n)}$, flag f indicating whether the PrimaryRational, Jordan, or Rational form is desired.

Output:
$F \in K^{(n\times n)}$ and $T \in K^{(n\times n)}$, such that F is the desired form and T is a non-singular matrix such that $TXT^{-1} = F$. If $f =$ PrimaryRational or $f =$ Jordan, the algorithm also returns $k \in \mathbb{Z}^+$, $p_1(x), \ldots, p_k(x) \in K[x]$, and $e_1, \ldots, e_k \in \mathbb{Z}^+$ such that $p_i(x)$ is irreducible for $1 \le i \le k$ and the $p_i(x)^{e_i}$ are the primary invariant factors of X and the blocks of F correspond to these polynomial powers. Otherwise ($f =$ Rational), the algorithm also returns $r \in \mathbb{Z}^+$, and $f_1(x), \ldots, f_r(x) \in K[x]$ such that the $f_i(x)$ are the invariant factors of X (so $f_1(x) \mid f_2(x) \mid \cdots \mid f_r(x)$), and the blocks of F correspond to these polynomials.

Method:
Apply algorithm Decompose to V and X to obtain a decomposition of V into a direct sum of primary spaces V_i such that the minimal polynomial of each V_i is a power of an irreducible polynomial and a cyclic generator for each V_i *may* be known.

Then for each space V_i such that a cyclic generator of V_i is *not* known, apply algorithm SplitPrimary to $W = V_i$ and X (with the appropriate irreducible polynomial $f(x)$ and its multiplicity q) to obtain a cyclic decomposition of V_i into primary spaces and replace V_i by the direct sum of the new cyclic primary spaces obtained.

Thus a decomposition of V into cyclic primary spaces is finally found. Sort the spaces with respect to the minimal polynomials so that spaces whose minimal polynomials are the power of a common irreducible polynomial are consecutive and such spaces are also ordered with respect to the multiplicity of the power. Thus we have $k \in \mathbb{Z}^+$, $v_i \in V$,

$p_i(x) \in K[x]$, and $e_i \in \mathbb{Z}^+$ for $1 \leq i \leq k$ with

$$V = \mathrm{Orb}_X(v_1) \oplus \cdots \oplus \mathrm{Orb}_X(v_k),$$

$\mathrm{Min}_X(v_i) = p_i(x)^{e_i}$, and $p_i(x)$ irreducible for $1 \leq i \leq k$.

If $f = \mathsf{PrimaryRational}$, let

$$F = \begin{pmatrix} \mathrm{C}(p_1(x)^{e_1}) & 0 & \dots & 0 \\ 0 & \mathrm{C}(p_2(x)^{e_2}) & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & \mathrm{C}(p_k(x)^{e_k}) \end{pmatrix}, \qquad T = \begin{pmatrix} \mathrm{TR}_X(v_1, \partial p_1(x)^{e_1}) \\ \mathrm{TR}_X(v_2, \partial p_2(x)^{e_2}) \\ \vdots \\ \mathrm{TR}_X(v_k, \partial p_k(x)^{e_k}) \end{pmatrix},$$

and output F, T, k, $p_i(x)$ and e_i for $1 \leq i \leq k$.

If $f = \mathsf{Jordan}$, let

$$F = \begin{pmatrix} \mathrm{J}(p_1(x), e_1) & 0 & \dots & 0 \\ 0 & \mathrm{J}(p_2(x), e_2) & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & \mathrm{J}(p_k(x), e_k) \end{pmatrix}, \qquad T = \begin{pmatrix} \mathrm{TJ}_X(v_1, p_1(x), e_1) \\ \mathrm{TJ}_X(v_2, p_2(x), e_2) \\ \vdots \\ \mathrm{TJ}_X(v_k, p_k(x), e_k) \end{pmatrix},$$

and output F, T, k, $p_i(x)$ and e_i for $1 \leq i \leq k$.

If $f = \mathsf{Rational}$, apply algorithm CombineSpaces to the cyclic primary spaces. The algorithm is applicable because of the way the spaces have been sorted. We thus obtain $r \in \mathbb{Z}^+$, $w_i \in V$ and $f_i(x) \in K[x]$ for $1 \leq i \leq r$ such that

$$V = \mathrm{Orb}_X(w_1) \oplus \cdots \oplus \mathrm{Orb}_X(w_r),$$

$\mathrm{Min}_X(w_i) = f_i(x)$ for $1 \leq i \leq r$, and

$$f_1(x) \mid f_2(x) \mid \cdots \mid f_r(x).$$

Let

$$F = \begin{pmatrix} \mathrm{C}(f_1(x)) & 0 & \dots & 0 \\ 0 & \mathrm{C}(f_2(x)) & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & \mathrm{C}(f_r(x)) \end{pmatrix}, \qquad T = \begin{pmatrix} \mathrm{TR}_X(w_1, \partial f_1(x)) \\ \mathrm{TR}_X(w_2, \partial f_2(x)) \\ \vdots \\ \mathrm{TR}_X(w_r, \partial f_r(x)) \end{pmatrix},$$

and output F, T, r, and $f_i(x)$ for $1 \leq i \leq k$.

Proof of Correctness. The correctness of the algorithm follows from the proofs of correctness of the algorithms in the previous sections. It is easy to show that the transformation matrix T in each case conjugates X to the canonical form F because of the properties of the row vectors generating each component of T. □

7. Analysis

We now give a brief analysis of the performance of the algorithm. We suppose that n, the dimension of the vector space $V = K^{(n)}$, indicates the size of the input. The algorithm has a theoretical complexity of $\mathrm{O}(n^4)$ field operations. However, this theoretical complexity does not indicate much about the performance of the algorithm.

We assume that multiplying a vector v from V by the matrix $X \in K^{(n \times n)}$ takes $\mathrm{O}(n^2)$

field operations. The algorithm MinOrb clearly takes $O(n^3)$ field operations since it performs $O(n)$ vector multiplications.

The algorithm Decompose takes $O(n^4)$ field operations since the body of the main loop at label `A` is executed $O(n)$ times and the body is dominated by the call to MinOrb to compute the minimal polynomials and orbits of the vectors considered in it.

The algorithm SplitPrimary is improved in implementation by not using the matrix X but a smaller matrix Y (of size $m \times m$ where m is the dimension of W) which represents the reduced action of X on W. Clearly it takes $O(n^3)$ field operations to compute Y. Algorithm SplitPrimary then takes $O(m^4)$ field operations where m is the dimension of the given subspace, since it is dominated by the evaluation of Y (standing for X) in the polynomial $f(x)$ to obtain N, and the calculation of (bases for) the kernels of the powers of N, both of which take $O(m^4)$ field operations. Now the main algorithm calls SplitPrimary possibly several times on, say, r subspaces of dimensions $m_1, \ldots, m_r$. Then the total number of field operations used in all the calls of SplitPrimary is $O((m_1^4 + m_2^4 + \cdots + m_r^4) + rn^3)$, which can be easily shown to be less than or equal to $O(n^4)$ since $m_1 + m_2 + \cdots + m_r \leq n$ and $r \leq n$.

The algorithm CombineSpaces is dominated by the computation of the polynomial powers $p_i(x)^{e_{i,j}}$ for various i and j each of which takes $O(n^2 \log n)$ field operations. But there are at most $O(n)$ such powers evaluated so the algorithm takes $O(n^3 \log n)$ field operations.

Finally, the main algorithm thus takes theoretically $O(n^4)$ field operations since algorithm Decompose is called once, algorithm CombineSpaces is called once or not at all, and the total running time of all calls of algorithm SplitPrimary was shown above to be $O(n^4)$.

This theoretical estimate does not say much about the performance of the algorithm. In the practical implementation of the algorithm in Magma, the running time of the algorithm is invariably dominated by far by the algorithm Decompose. In turn, the running time of that algorithm is vastly affected by the structure of the primary invariant factors of the given matrix. If this structure does not include many repeated factors but a diversity of factors, then the main loop in that algorithm is usually only executed a few times. This situation arises very often.

The algorithm SplitPrimary actually is called rarely and when it is called it is usually with input of trivial size. If there are not many repeated factors in the primary invariant factors, then algorithm Decompose usually finds cyclic generators for all the subspaces produced by it.

When the main algorithm is called, any of the canonical form, the transformation matrix, or the (primary) invariant factors can be omitted. For example, one important application of the algorithm is to test two matrices for similarity. Computing the primary invariant factors of each of the matrices gives the answer. If they are similar and a matrix which conjugates one matrix to the other is desired, that can be easily derived from the transformation matrices computed by the algorithm for each of the input matrices. In either case, the actual canonical forms are not necessary.

8. Example

In this section we present an example of the use of the Magma implementation of the algorithm.

We consider a 10×10 matrix X over the field $\mathbb{Q}$ of rational numbers. We first print the matrix X.

```
> print X;
[-23  19  -9 -75  34   9  56  15 -34  -9]
[ -2   2  -1  -6   3   1   4   2  -3   0]
[  4  -4   3  10  -5  -1  -6  -4   5   1]
[ -2   2  -1  -5   3   1   3   2  -3   0]
[  0   0   0   0   2   0   0   0   0   0]
[ 12 -12   6  33 -18  -4 -18 -12  18   0]
[ -1  -3   0   2   1   0   1   1   2   1]
[-26  22 -10 -83  36  10  61  18 -39 -10]
[ -1  -3   0   1   1   0   2   1   2   0]
[  8 -12   4  27 -12  -4 -12  -7  15   0]
```

Next we compute the Primary Rational form of X. We use the MAGMA function `PrimaryRationalForm` which has 3 return values: P, the Primary Rational form of X; PT, the transformation matrix such that $PT * X * PT^{-1} = P$; and PF, the primary invariant factors of X.

```
> P, PT, PF := PrimaryRationalForm(X);
> print P;
[  2   0   0   0   0   0   0   0   0   0]
[  0   2   0   0   0   0   0   0   0   0]
[  0   0   0   1   0   0   0   0   0   0]
[  0   0  -4   4   0   0   0   0   0   0]
[  0   0   0   0   0   1   0   0   0   0]
[  0   0   0   0   3  -4   0   0   0   0]
[  0   0   0   0   0   0   0   1   0   0]
[  0   0   0   0   0   0   0   0   1   0]
[  0   0   0   0   0   0   0   0   0   1]
[  0   0   0   0   0   0  -9  24 -10  -8]
> print PT*X*PT^-1 eq P;
true
> print PF;
[
    <x - 2, 1>,
    <x - 2, 1>,
    <x - 2, 2>,
    <x^2 + 4*x - 3, 1>,
    <x^2 + 4*x - 3, 2>
]
```

Notice that the block associated with $\langle x-2, 2\rangle$ is the companion matrix of $(x-2)^2$—the block is not simplified any further. The same fact holds for the block associated with $\langle x^2 + 4 * x - 3, 2\rangle$.

Next we compute the *generalized* Jordan form of X. (X cannot have a true Jordan form because the characteristic polynomial of X does not factorize completely into linear

factors over $\mathbb{Q}$.) This time we use the MAGMA function `JordanForm` which has 3 return values: J, the generalized Jordan form of X; JT, the transformation matrix such that $JT * X * JT^{-1} = J$; and JF, the primary invariant factors of X.

```
> J, JT, JF := JordanForm(X);
> print J;
[ 2  0  0  0  0  0  0  0  0  0]
[ 0  2  0  0  0  0  0  0  0  0]
[ 0  0  2  1  0  0  0  0  0  0]
[ 0  0  0  2  0  0  0  0  0  0]
[ 0  0  0  0  0  1  0  0  0  0]
[ 0  0  0  0  3 -4  0  0  0  0]
[ 0  0  0  0  0  0  0  1  0  0]
[ 0  0  0  0  0  0  3 -4  1  0]
[ 0  0  0  0  0  0  0  0  0  1]
[ 0  0  0  0  0  0  0  0  3 -4]
> print JT*X*JT^-1 eq J;
true
> print JF eq PF;
true
```

Notice that the primary invariant factors JF are the same as the primary invariant factors PF returned by `PrimaryRationalForm`. Notice that this time the block associated with $\langle x-2, 2\rangle$ is the appropriate Jordan block—the block is "simpler" than the corresponding block in the Primary Rational form. Similarly, the block associated with $\langle x^2+4*x-3, 2\rangle$ is a diagonal joining of two copies of the companion matrix of $x^2+4*x-3$ with a single 1 above the diagonal.

Finally, we compute the Rational form of X. This time we use the MAGMA function `RationalForm` which has 3 return values: F, the Rational form of X; RT, the transformation matrix such that $RT * X * RT^{-1} = R$; and RF, the invariant factors of X.

```
> R, RT, RF := RationalForm(X);
> print R;
[   2    0    0    0    0    0    0    0    0    0]
[   0    0    1    0    0    0    0    0    0    0]
[   0    0    0    1    0    0    0    0    0    0]
[   0   -6   11   -2    0    0    0    0    0    0]
[   0    0    0    0    0    1    0    0    0    0]
[   0    0    0    0    0    0    1    0    0    0]
[   0    0    0    0    0    0    0    1    0    0]
[   0    0    0    0    0    0    0    0    1    0]
[   0    0    0    0    0    0    0    0    0    1]
[   0    0    0    0  -36  132 -145   32   18   -4]
> print RT*X*RT^-1 eq R;
true
> print RF;
[
    x - 2,
```

```
    x^3 + 2*x^2 - 11*x + 6,
    x^6 + 4*x^5 - 18*x^4 - 32*x^3 + 145*x^2 - 132*x + 36
]
```

Notice that this time the blocks are as big as possible and the polynomials in RF satisfy the divisibility condition.

To give also an indication of the efficiency of the algorithm, we give here some simple timings of the MAGMA implementation of it, running on a Sun 4/670MP server. The generalized Jordan form of random matrices of differing degrees over the finite field with two elements were computed. Because the matrices were random, the structure of the (primary) invariant factors was rather simple each time (i.e., there were not many repeated factors). The timings were: degree 200: 2.3 seconds; degree 400: 4.8 seconds; degree 600: 29.4 seconds; degree 800: 74.4 seconds; degree 1000: 207.8 seconds.

Acknowledgements

The author would like to thank Charles Leedham-Green for suggesting the general ideas of computing minimal polynomials and orbits by the "spinning a vector" method. The author would also like to particularly thank Bob Howlett for ideas which motivated the algorithms in sections 4 and 5.

References

Bosma, W., Cannon, J., Playoust, C. (1995). The Magma algebra system I: the user language. *J. Symbolic Comput.* **24**, 235–265.

Dixon, J.D., Poland, J.C., Pressman, I.S., Ribes, L. (1990). The shuffle algorithm and Jordan blocks. *Linear Algebra and its Applications* **142**, 159–165.

Faddeeva, V.N. (1959). *Computational Methods of Linear Algebra*. New York: Dover.

Hartley, B., Hawkes, T.O. (1970). *Rings, Modules and Linear Algebra*. London: Chapman and Hall.

Howell, J. (1973). An Algorithm for the Exact Reduction of a Matrix to Frobenius Form Using Modular Arithmetic, I + II. *Math. Comp.* **27**, 887–920.

Mathieu, M.-H., Ford, D. (1990). On p-adic Computation of the Rational Form of a Matrix. *J. Symbolic Comput.* **10**, 453–464.

Originally received 15 September 1995
Accepted 31 July 1995